On Bounded Distance Decoding, Unique Shortest Vectors, and the Minimum Distance Problem

Vadim Lyubashevsky[1] * and Daniele Micciancio[2] **

[1] School of Computer Science, Tel Aviv University
Tel Aviv 69978, Israel.
vlyubash@cs.ucsd.edu.
[2] Computer Science and Engineering Department,
University of California at San Diego,
La Jolla, CA 92093, USA.
daniele@cs.ucsd.edu

Abstract. We prove the equivalence, up to a small polynomial approximation factor $\sqrt{n/\log n}$, of the lattice problems uSVP (unique Shortest Vector Problem), BDD (Bounded Distance Decoding) and GapSVP (the decision version of the Shortest Vector Problem). This resolves a long-standing open problem about the relationship between uSVP and the more standard GapSVP, as well the BDD problem commonly used in coding theory. The main cryptographic application of our work is the proof that the Ajtai-Dwork ([2]) and the Regev ([33]) cryptosystems, which were previously only known to be based on the hardness of uSVP, can be equivalently based on the hardness of worst-case GapSVP$_{O(n^{2.5})}$ and GapSVP$_{O(n^2)}$, respectively. Also, in the case of uSVP and BDD, our connection is very tight, establishing the equivalence (within a small constant approximation factor) between the two most central problems used in lattice based public key cryptography and coding theory.

1 Introduction

Lattice based cryptography is among the most compelling alternatives to traditional methods based on number theory. Ajtai's ground-breaking discovery that lattice problems exhibit a worst-case to average-case connection [1] immediately yielded one-way functions and collision resistant hash functions based on the worst-case hardness of several lattice approximation problems, and prompted researchers to investigate the construction of more complex cryptographic primitives (most notably public key encryption) based on lattices. The first cryptosystem that was based on the worst-case hardness of lattice problems was

* Supported by the Israel Science Foundation and by a European Research Council (ERC) Starting Grant

** Supported in part by NSF grant CCF-0634909. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

the Ajtai-Dwork cryptosystem [2]. The security of this system was based on the worst-case hardness of the approximate "unique" Shortest Vector Problem uSVP$_{O(n^8)}$ (in uSVP$_\gamma$, we are asked to find the shortest vector in a lattice in which the shortest vector is guaranteed to be at least γ times smaller than the next shortest non-parallel lattice vector). This was followed by an improvement to their cryptosystem [11], and the currently best version of it is based on the hardness of uSVP$_{O(n^2)}$. In a later work, Regev built a different cryptosystem based on worst-case uSVP$_{O(n^{1.5})}$ [33]. But while other cryptographic primitives could be built on the hardness of the more general, and better understood from a complexity-theoretic point of view, shortest vector problem on general lattices (in its decision variant, GapSVP), cryptosystems seemed to require the hardness of the potentially easier uSVP lattices. So it was a major open problem as to whether lattice-based cryptosystems could be based on the hardness of problems on general lattices, and GapSVP in particular. What made the problem even more interesting was that simpler cryptographic primitives from "minicrypt"[3] such as one-way functions [1], collision-resistant hash functions [1,10,29] identification schemes [30,25,18] and signature schemes [26,8] could be based on the worst-case hardness of GapSVP.

A breakthrough in the design of lattice-based cryptosystems, in the sense of deviating from uSVP, came when Regev built a cryptosystem which was actually based on GapSVP (as well as some other standard lattice problems), but the assumption was that approximating GapSVP was hard even by quantum algorithms [35]. Another breakthrough came just recently, when Peikert finally constructed a cryptosystem that is based on the hardness of GapSVP under classical reductions [32]. Of course, a different way of obtaining cryptosystems with security based on GapSVP would be to establish a relation between GapSVP and uSVP, and this is precisely what we do in this paper.

On the practical front, about at the same time as Ajtai's discovery [1], two cryptosystems were proposed (GGH [12] and NTRU [15]), which, while lacking a security proof from worst-case lattice assumptions, are intuitively very appealing. These cryptosystems rest on the conjectured average-case hardness of the bounded distance decoding problem (BDD), which can be considered a special version of the closest vector problem, very much like uSVP is a special version of the shortest vector problem. Additionally, Regev's cryptosystem [35] whose security is based on the worst-case hardness of quantum GapSVP is equivalently based on an average-case version of classical BDD (used in [35] under the name "Learning with Errors" problem.) So, the average-case BDD problem seems quite a natural problem to consider in the setting of lattice based public key encryption.

Our contribution. In this paper, we prove the equivalence, up to a factor of $\sqrt{n/\log n}$, of the GapSVP, BDD, and uSVP problems. In particular, we prove

[3] Minicrypt [16] consists of all cryptographic primitives that can be derived from one-way functions, or more generally, exist relative to a random oracle. Collision resistant hash functions are not known to be reducible to one-way functions, but still exist relative to a random oracle, so they can be included in minicrypt.

Cryptosystem	GapSVP Approximation Factor	Message Expansion
Ajtai-Dwork [2]	$\tilde{\mathbf{O}}(\mathbf{n^{2.5}})$	$O(n^2)$
Regev [33]	$\tilde{\mathbf{O}}(\mathbf{n^{2}})$	$O(n)$
Peikert [32]	$\tilde{O}(n^2)$	$O(\log n)$

Fig. 1. Cryptosystems based on worst-case GapSVP$_\gamma$. The results in bold-face are consequences of the current work.

that for any $\gamma \geq 1$, there is a reduction from BDD$_{1/2\gamma}$ to uSVP$_\gamma$, and for any polynomially-bounded γ, there is a reduction from uSVP$_\gamma$ to BDD$_{1/\gamma}$. (We remark that the BDD$_\alpha$ problem is easier for *smaller* values of the factor α, while uSVP$_\gamma$ is easier for *larger* values of γ. For a formal definition of the problems, see the next section.) So, the problems uSVP$_\gamma$ and BDD$_{1/\gamma}$ are essentially equivalent under polynomial time reduction that preserve the approximation factor up to a small constant $\gamma/\gamma' \leq 2$. We also show reductions from uSVP$_\gamma$ to GapSVP$_\gamma$, and from GapSVP$_\gamma$ to BDD$_{\frac{1}{\gamma}\sqrt{n/\log n}}$ (for any $\gamma > 2\sqrt{n/\log n}$).

So, in summary, all three problems uSVP$_\gamma$, BDD$_{1/\gamma}$ and GapSVP$_\gamma$ are equivalent up to polynomial approximation factors, and all currently known lattice based public key cryptosystems with classical worst-case security guarantees [2,33,32] are qualitatively equivalent. In particular, our results imply that the Ajtai-Dwork [2] and the Regev [33] cryptosystems are based on the hardness of GapSVP$_{\tilde{O}(n^{2.5})}$ and GapSVP$_{\tilde{O}(n^2)}$ respectively. And since Peikert's recent cryptosystem [32] is also based on the hardness of the GapSVP$_{\tilde{O}(n^2)}$, the only major quantitative difference between the three cryptosystems is that Peikert's has a smaller message expansion factor (see Figure 1 and also [32] for more details).

When it comes to the practical GGH [12] and NTRU [15] cryptosystems, we cannot formally draw any implications from our findings, because ours are worst-case to worst-case reductions, and the GGH and NTRU cryptosystems lack security proofs from worst-case problems. Still, our results show that the (average-case) BDD lattice problems underlying GGH and NTRU, and those used in more theoretical constructions, have much more in common than previously thought.

In addition to cryptographic applications, the uSVP problem also found applicability in areas of learning theory and quantum computation. Klivans and Sherstov showed that a polynomial-time algorithm PAC-learning the intersection of n^ϵ half-spaces implies a polynomial-time algorithm for solving uSVP [20]. Regev showed that a solution to the dihedral coset problem would imply a quantum algorithm for uSVP [34]. Our work implies that the two problems above are based on the more well-studied GapSVP problem. This seems especially important for Regev's result since there was very little prior evidence that uSVP was hard for quantum computers.

1.1 Previous Work

There has been a lot of work in establishing relationships between various lattice problems. In fact, while our central cryptographic result (that the Ajtai-Dwork and the Regev cryptosystems are based on the hardness of GapSVP) is new, the components that comprise it are very much based on prior work.

Proving the reduction from GapSVP to uSVP can be broken down into two separate reductions. In section 4, we give a reduction from BDD to uSVP and in section 7 we give a reduction from GapSVP to BDD. The BDD to uSVP reduction uses an idea that dates back to at least the classic result of Lagarias and Olyzsko [22] where random low-density subset sum instances are converted to lattices with a unique shortest vector. This same idea has subsequently been used in various guises in reductions [17,6] as well as in heuristic attacks on cryptographic primitives [31].

The reduction from GapSVP to BDD is already implicit in the recent work of Peikert [32]. And in fact, almost the same idea was already used in the work of Goldreich and Goldwasser [9] where it was proved that GapSVP (and other lattice problems) are in the complexity class coAM. In that work, an all-powerful prover was able to convince a polynomially-bounded verifier that the length of the shortest vector of the lattice is large. The GapSVP to BDD reduction is obtained by realizing that the all-powerful prover in the coAM protocol can simply be substituted with a BDD oracle.

The other two reductions presented in our work are also related to some previous works. The reduction from uSVP to BDD in section 5 uses some ideas from the SVP to CVP reduction of Goldreich, et al. [13]. The reduction from uSVP to GapSVP is based on Regev's reduction from the decision to the search version of uSVP [33], but our proof is somewhat simpler and tighter.

1.2 Discussion and Open Problems

As mentioned earlier, one of the separations between the lattice-based "minicrypt" primitives and lattice-based public key cryptosystems was that the former could be based on the hardness of classical GapSVP, whereas the latter could not. But our work, as well as the recent work of Peikert [32], shows that there are cryptosystems based on the worst-case hardness of the shortest vector problem in its decision version. Nevertheless, there still seems to be a difference in the types of problems that "minicrypt" primitives can be based on and the hardness assumptions needed for public-key cryptosystems. The aforementioned "minicrypt" primitives [1,10,29,30,25,18,26,8] can all be based on a standard lattice *search* problem SIVP, in addition to GapSVP. We remark that, up to a polynomial loss in the approximation factor, GapSVP, uSVP and BDD can be reduced to SIVP.[4] Moreover the "quantum step" of [35] gives a *quantum*

[4] This can be done in a variety of ways. For example, one can first reduce $\text{GapSVP}_{n\gamma}$ to GapSIVP_{γ} using transference theorems, and then use a trivial reduction from GapSIVP_{γ} to SIVP_{γ}.

reduction from $\mathrm{SIVP}_{O(n\gamma)}$ to $\mathrm{BDD}_{1/\gamma}$. So, under quantum reductions, all lattice problems uSVP, BDD, GapSVP, SIVP are qualitatively equivalent, up to polynomial approximation factors. However, there is no known classic polynomial time reduction from SIVP to any of uSVP, BDD, GapSVP (except in trivial cases). We also remark that the two most famous lattice problems, SVP and CVP, are equivalent under polynomial time reduction up to polynomial approximation factors [17], and there is an approximation preserving reduction from SIVP to CVP [27]. However, there is no known reduction in the opposite direction, from SVP or CVP to SIVP. So once again, this raises the question of whether lattice-based public key cryptosystems require qualitatively stronger assumptions than simpler cryptographic primitives (e.g., quantum hardness of SIVP, rather than just classic hardness), and whether cryptography in general can be based on the worst-case hardness of SVP or CVP in their search version.

It is interesting to point out that even though the cryptosystems described in [2,33,32] are all based on the hardness of GapSVP, the construction of Peikert's cryptosystem is quite different from the other two. The Regev and Ajtai-Dwork cryptosystems were based directly on the uSVP problem, while Peikert's cryptosystem is actually quite similar to the other Regev cryptosystem [35] whose hardness is based on BDD. At this point, cryptosystems based on BDD are more efficient since their message expansion factor is smaller (see Figure 1), but perhaps the connection between GapSVP, BDD, and uSVP demonstrated in this work can be somehow exploited in order to combine the two seemingly distinct techniques for cryptosystem construction and build one that is even more efficient and still based on the hardness of GapSVP.

Another outstanding question on the complexity of lattice problems is whether the search and length estimation/decision versions of the shortest vector problem are computationally equivalent. A search to decision reduction for the approximate SVP would immediately imply the equivalence (up to polynomial factors) of all lattice problems uSVP, BDD, GapSVP, SVP, CVP, SIVP considered in cryptography.

There are also many questions about the relationship between lattice problems that are raised directly from our work. One such problem is whether the reductions in sections 5 and 6 can be extended to approximation factors that are not restricted to being polynomial. Another problem is to figure out whether the small gap that we have in the connection between BDD and uSVP can be closed. At this point, we have the reduction $\mathrm{uSVP}_\gamma \leq \mathrm{BDD}_{1/\gamma} \leq \mathrm{uSVP}_{\gamma/2}$, which is loose by a factor of 2. We believe that there are three (mutually exclusive) possibilities for possible improvements of this result. It might be possible to show that:

1. $\mathrm{uSVP}_{\gamma/2} \leq \mathrm{BDD}_{1/\gamma}$ or
2. $\mathrm{BDD}_{1/\gamma} \leq \mathrm{uSVP}_\gamma$ or
3. $\mathrm{uSVP}_\gamma \leq \mathrm{BDD}_{\sqrt{2}/\gamma} \leq \mathrm{uSVP}_\gamma$

This open problem also has an intriguing connection with the computational complexity of uSVP_γ. Unlike SVP_γ, which is known to be NP-hard for any constant γ, uSVP_γ is only NP-hard for $\gamma = 1 + 2^{-n^c}$ for some constant c [21].

So proving the NP-hardness of uSVP$_\gamma$ for larger factors is a very interesting open problem. One possibility for doing so would be to prove item (2) above. Combining this with the result of [24] that states that BDD$_\gamma$ is NP-hard for $\gamma > 1/\sqrt{2}$, we would obtain that uSVP$_\gamma$ is NP-hard for $\gamma = \sqrt{2}$.

The possibility of somehow using the reduction from GapSVP to uSVP in order to prove NP-hardness of uSVP is also intriguing. At this point, the reduction requires the γ in GapSVP$_\gamma$ to be at least $\sqrt{n/\log n}$, and the GapSVP$_\gamma$ problem is not NP-hard for such parameters unless the polynomial-time hierarchy collapses. While there seem to be some technical roadblocks for reducing this requirement, it is not entirely clear that this should not be possible.

1.3 Organization of the Paper

Those readers interested mainly in the reduction from GapSVP to uSVP (which implies that the security of the Ajtai-Dwork and Regev cryptosystems is based on worst-case GapSVP) can simply read sections 4 and 7 for the proofs of BDD $\leq$ uSVP and GapSVP $\leq$ BDD respectively. Section 4 uses a result from section 3 in order to strengthen the reduction a bit, but this can be safely skipped.

In order to establish the equality of the three problems, we also need to prove that uSVP $\leq$ BDD and uSVP $\leq$ GapSVP. This is done in sections 5 and 6 respectively.

2 Preliminaries

An n-dimensional *lattice* is a discrete additive subgroup of $\mathbb{R}^n$. A set of linearly independent vectors that generates a lattice is called a basis, and we will denote it as an $n \times m$ matrix $\mathbf{B}$ whose m columns $\mathbf{b}_i$ are the generating vectors. The lattice generated by the basis $\mathbf{B}$ will be written as $\mathcal{L}(\mathbf{B})$. The *span* of a basis $\mathbf{B}$, denoted $Span(\mathbf{B})$, is the collection of all points $\mathbf{By}$ where $\mathbf{y} \in \mathbb{R}^m$. The *fundamental parallelepiped* of an $n \times m$ basis $\mathbf{B}$, written as $\mathcal{P}(\mathbf{B})$, is defined as the collection of all points that can be written as $\mathbf{By}$ where $\mathbf{y} \in [0,1)^m$. Every point $\mathbf{s} \in \mathbb{R}^n$ has a unique associated point $\mathbf{t}$ inside $\mathcal{P}(\mathbf{B})$ such that $\mathbf{s} = \mathbf{t}$ in the quotient group $Span(\mathbf{B})/\mathcal{L}(\mathbf{B})$. This point is denoted $\mathbf{t} = \mathbf{s} \bmod \mathbf{B}$ and can be computed from $\mathbf{s}$ in polynomial time. For any point $\mathbf{t}$, in $\mathbb{R}^n$ and any lattice $\mathcal{L}(\mathbf{B})$, the distance of $\mathbf{t}$ to the lattice is written as $\text{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$.

For any point $\mathbf{t} \in \mathbb{R}^n$ and $r \in \mathbb{R}$, let $\mathcal{B}(\mathbf{t}, r)$ denote a ball of radius r centered at $\mathbf{t}$. The *shortest vector* of a lattice $\mathcal{L}(\mathbf{B})$ is the non-zero vector in $\mathcal{L}(\mathbf{B})$ with the smallest ℓ_2 norm. The length of the shortest vector, referred to as the *minimum distance*, of $\mathcal{L}(\mathbf{B})$ is denoted by $\lambda_1(\mathcal{L}(\mathbf{B}))$ (or $\lambda_1(\mathbf{B})$ for short). The notion of minimum distance can be generalized to define the i^{th} successive minimum $\lambda_i(\mathbf{B})$ as the smallest radius r such that $\mathcal{B}(\mathbf{0}, r)$ contains i linearly independent lattice points. The determinant of a lattice $\mathcal{L}(\mathbf{B})$ is defined as $\sqrt{det(\mathbf{B}^T\mathbf{B})}$. When $\mathbf{B}$ is a full-rank lattice, the previous definition becomes just $|det(\mathbf{B})|$. Lattices $\mathcal{L}(\mathbf{B})$ and $\mathcal{L}(\mathbf{D})$ are called *dual* if $\mathcal{L}(\mathbf{D}) = \{\mathbf{y} : \forall \mathbf{v} \in \mathcal{L}(\mathbf{B}), \mathbf{y} \cdot \mathbf{v} \in \mathbb{Z}\}$. If $\mathcal{L}(\mathbf{B})$ and

$\mathcal{L}(\mathbf{D})$ are duals, then $det(\mathcal{L}(\mathbf{B})) = det(\mathcal{L}(\mathbf{D}))^{-1}$. Minkowski's theorem states that for any n-dimensional lattice $\mathcal{L}(\mathbf{B})$, $\lambda_1(\mathcal{L}(\mathbf{B})) \leq \sqrt{n} \cdot det(\mathcal{L}(\mathbf{B}))^{1/n}$. For additional information about lattices, please refer to [28].

2.1 GapSVP

Possibly the most well-known lattice problem is the Shortest Vector Problem (SVP). It comes in both decisional and search versions, but in this paper we are only interested in the decision version. (The decision version of the problem is sometimes referred to as the *Minimum Distance Problem*). The approximation version of decisional SVP can be defined as a "gap" problem GapSVP$_\gamma$. In the GapSVP$_\gamma$ problem, we are given a basis $\mathbf{B}$ and a real number d, and are required to return YES if $\lambda_1(\mathcal{L}(\mathbf{B})) \leq d$, and return NO if $\lambda_1(\mathcal{L}(\mathbf{B})) > \gamma d$. If $\lambda_1(\mathcal{L}(\mathbf{B}))$ falls between d and γd, we can return anything. The GapSVP$_\gamma$ problem is NP-hard for any constant γ [19,14]. The fastest algorithm for solving GapSVP$_\gamma$ for $1 \leq \gamma \leq poly(n)$ takes time $2^{O(n)}$ [3]. Using the LLL algorithm [23], it is possible to find a vector that has length at most $2^{n/2}\lambda_1(\mathbf{B})$ in polynomial time.

2.2 uSVP and BDD

We now give precise definitions for the other two lattice problems that are central to this work. We urge the reader to notice that while the minimum distance problem described in the previous section is a *decision* problem, the ones in this section are *search* problems.

Definition 1 (γ-unique Shortest Vector (uSVP$_\gamma$)). *Given a lattice* $\mathbf{B}$ *such that* $\lambda_2(\mathbf{B}) > \gamma\lambda_1(\mathbf{B})$*, find a nonzero vector* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *of length* $\lambda_1(\mathbf{B})$.

Definition 2 (α-Bounded Distance Decoding (BDD$_\alpha$)). *Given a lattice basis* $\mathbf{B}$ *and a vector* $\mathbf{t}$ *such that* $\operatorname{dist}(\mathbf{t}, \mathbf{B}) < \alpha\lambda_1(\mathbf{B})$*, find the lattice vector* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *closest to* $\mathbf{t}$.

The uSVP$_\gamma$ problem is known to be NP-hard when $\gamma = 1 + 2^{-n^c}$ for some constant c [21], and it's an outstanding open problem whether NP-hardness can be proved for larger γ. There has been some evidence to suggest that uSVP is easier than the search version of SVP [7], and that approximating the length of the shortest vector in lattices with a unique shortest vector may be easier than GapSVP in general lattices [5].

The BDD$_\alpha$ problem has been shown to be NP-hard for $\alpha > 1/\sqrt{2}$ [24] and it is an open problem whether it's hard for smaller α. We would just like to draw the reader's attention to the fact that the BDD$_\alpha$ problem becomes *harder* as α becomes larger, while the uSVP$_\gamma$ problem becomes *easier* as γ increases. Sometimes uSVP$_\gamma$ and BDD$_\alpha$ are defined in a more relaxed way, as follows:

Definition 3 (uSVP$'_\gamma$). *Given a lattice* $\mathbf{B}$ *such that* $\lambda_2(\mathbf{B}) > \gamma\lambda_1(\mathbf{B})$*, find a nonzero vector* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *of length* $\|\mathbf{v}\| \leq \gamma\lambda_1(\mathbf{B})$.

Definition 4 (BDD'_α)**.** *Given a lattice basis* $\mathbf{B}$ *and a vector* $\mathbf{t}$ *such that* $\mathrm{dist}(\mathbf{t}, \mathbf{B}) < \alpha\lambda_1(\mathbf{B})$*, find a lattice vector* $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ *such that* $\|\mathbf{v} - \mathbf{t}\| < \alpha\lambda_1(\mathbf{B})$*.*

However, these relaxed variants are not any easier to solve than uSVP_γ and BDD_α as defined in this paper.

Lemma 1. *For any* $\gamma \geq 1$*, the problems* uSVP_γ *and* uSVP'_γ *are equivalent under polynomial time reductions.*

Proof. Clearly, uSVP'_γ reduces to uSVP_γ because any solution to uSVP_γ instance $\mathbf{B}$ is also a relaxed solution to $\mathbf{B}$ as a uSVP'_γ instance. In the other direction, let $\mathbf{B}$ be a uSVP_γ instance. Using a uSVP'_γ oracle we can find a nonzero lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ of length $\|\mathbf{v}\| \leq \gamma\lambda_1(\mathbf{B})$. Using the uSVP restriction $\gamma\lambda_1 < \lambda_2$, we get that the shortest nonzero vector in $\mathcal{L}(\mathbf{B})$ must be of the form $c\mathbf{v}$ (for $c \in \mathbb{R}$), and can be easily found solving a 1-dimensional SVP instance $\mathcal{L}(\mathbf{B}) \cap \mathbf{v}\mathbb{R}$. □

The equivalence between BDD_α and BDD'_α is a bit trickier.

Lemma 2. *For any* $\alpha \geq 1$*, the problems* BDD_α *and* BDD'_α *are equivalent under polynomial time reductions.*

Proof. As in the previous theorem, the reduction from BDD'_α to BDD_α is trivial. Reducing BDD_α to BDD'_α is also trivial when $\alpha \leq 1/2$, because there is at most one lattice point within distance $\lambda_1(\mathbf{B})/2$ from any target. When $\alpha > 1/2$ the reduction is not as trivial because the BDD'_α oracle may return one of several lattice points, at distance from the target ranging from $\mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$ to $\alpha\lambda_1(\mathbf{B})$. This technical problem can be easily solved as follows. Let $(\mathbf{B}, \mathbf{t})$ be a BDD_α instance, and assume without loss of generality that $\mathbf{B}$ and $\mathbf{t}$ have integer entries. Consider the BDD'_α instance $(\mathbf{B}', \mathbf{t}')$ where

$$\mathbf{B}' = \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0}^T & d/\alpha \end{bmatrix} \qquad \mathbf{t}' = \begin{bmatrix} \mathbf{t} \\ 0 \end{bmatrix}$$

for some $d > 0$. Notice that we still have $\mathrm{dist}(\mathbf{t}', \mathcal{L}(\mathbf{B}')) = \mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$. In fact, the extra coordinate and basis vector in $\mathbf{B}'$ have the only effect of reducing the length of the shortest vector in the lattice to $\lambda_1(\mathbf{B}') = \min(\lambda_1(\mathbf{B}), d/\alpha)$. Let $\mu = \mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$. Then using lattice reduction algorithms (see Lemma 3) we can efficiently compute a lattice point $\mathbf{v}$ at distance $d_0 = \|\mathbf{v} - \mathbf{t}\| \in [\mu, 2^n\mu]$ from $\mathbf{t}$. If $d_0 = \mu$, then we have found the closest lattice point. So, assume $d_0 > \mu$. Notice that when $d = d_0$, the instance $(\mathbf{B}', t')$ satisfies

$$\alpha\lambda(\mathbf{B}') = \min\{d, \alpha\lambda(\mathbf{B})\} > \mu.$$

So, on input $(\mathbf{B}', \mathbf{t}')$, the BDD'_α oracle returns a lattice point $\mathbf{B}'\mathbf{z}'$ such that $\|\mathbf{B}'\mathbf{z}' - \mathbf{t}'\| < \alpha\lambda(\mathbf{B}') \leq d$. On the other hand, if $d = d_0/2^n$, then for any lattice point $\mathbf{B}'\mathbf{z}'$, we have $\|\mathbf{B}'\mathbf{z}' - \mathbf{t}'\| \geq \mu \geq d$. Using binary search, we can find a d_1 such that the BDD'_α oracle returns a lattice vector $\mathbf{B}'\mathbf{z}'$ such that $\|\mathbf{B}'\mathbf{z}' - \mathbf{t}'\| < d$

when $d = d_1\sqrt{1+1/d_0^2}$, but not when $d = d_1$. Without loss of generality, we can assume $\mathbf{z}' = [\mathbf{z}^T, 0]$ and $\|\mathbf{B}'\mathbf{z}' - \mathbf{t}'\| = \|\mathbf{B}\mathbf{z} - \mathbf{t}\|$.

We claim that $d_1 \leq \mu$, and therefore, since $\|\mathbf{B}\mathbf{z} - \mathbf{t}\|^2$ and μ^2 are integers,

$$\|\mathbf{B}'\mathbf{z}' - \mathbf{t}'\|^2 = \|\mathbf{B}\mathbf{z} - \mathbf{t}\| \leq \lfloor d_1^2(1 + d_0^{-2})\rfloor \leq \lfloor \mu^2 + (\mu/d_0)^2 \rfloor = \mu^2.$$

So, $\mathbf{B}\mathbf{z}$ is the lattice vector closest to $\mathbf{t}$.

In order to prove the claim, assume for contradiction that $d_1 > \mu$. Then, when $d = d_1$, $\alpha\lambda(\mathbf{B}') = \min(d_1, \alpha\lambda(\mathbf{B})) > \mu$. So, the BDD'_α promise is satisfied, and on input $(\mathbf{B}', \mathbf{t}')$, the BDD'_α oracle returns a lattice point $\mathbf{B}'\mathbf{z}'$ such that $\|\mathbf{B}'\mathbf{z}' - \mathbf{t}'\| < \alpha\lambda(\mathbf{B}') \leq d_1$. This is a contradiction because we had assumed the oracle returned a lattice point such that $\|\mathbf{B}'\mathbf{z}' - \mathbf{t}'\| \geq d_1$. □

2.3 Useful Lemmas

The first lemma, due to Babai [4], states that for any point in space, we can approximate the lattice point closest to it within a factor of 2^n.

Lemma 3. *There exists a polynomial-time algorithm that, given $\mathbf{t} \in \mathbb{R}^n$ and a lattice $\mathcal{L}(\mathbf{B})$, outputs a lattice vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}-\mathbf{t}\| \in [\mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})), \leq 2^n \mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))]$.*

The second lemma, due to Goldreich and Goldwasser [9], states that two spheres of large radii whose centers are relatively close to each other will have a relatively large (non-negligible) intersection.

Lemma 4. *Let $\mathbf{x}$ be a vector in $\mathbb{R}^n$ such that $\|\mathbf{x}\| \leq d$. If $\mathbf{s}$ is a point chosen uniformly at random from $\mathcal{B}(0, d\sqrt{n/\log n})$, then with probability $\delta > 1/n^c$ for some constant c, $\|\mathbf{s} - \mathbf{x}\| \leq d\sqrt{n/\log n}$.*

3 BDD Self-Reduction

In this section, we will show that there is a polynomial-time Cook reduction from solving BDD_α to the slightly easier problem $\mathrm{BDD}_{\alpha(1-1/n)^c}$ for any constant c. This reduction will be used to eliminate losses of small factors in our reductions that involve BDD.

Lemma 5. *For any $\alpha \geq 1$, there is a polynomial-time Cook reduction from BDD'_α to $\mathrm{BDD}'_{\alpha\sqrt{1-1/2n}}$.*

Proof. Let $(\mathbf{B}, \mathbf{t})$ be an instance of BDD'_α, and $\mathbf{y}$ be a vector in $\mathcal{L}(\mathbf{B})$ such that $\|\mathbf{t} - \mathbf{y}\| < \alpha\lambda_1(\mathbf{B})$. We do not know the actual distance $D = \|\mathbf{t} - \mathbf{y}\|$, but we can guess an approximation d such that

$$\frac{D}{2+\sqrt{2}} \leq d \leq \frac{D}{2-\sqrt{2}} \tag{1}$$

in polynomially many tries (this follows from Lemma 3 since we can approximate D to within a factor of 2^n). Consider the set

$$S = \left\{ \mathbf{t} - \frac{jd}{\sqrt{n}} \mathbf{u}_i : i \in \{1, \ldots, n\}, j \in \{-1, 1\} \right\}$$

where $\mathbf{u}_i$ is a vector with a 1 in the i^{th} position, and 0's everywhere else. We will show that this set contains a vector $\mathbf{t}'$ such that $\|\mathbf{t}' - \mathbf{y}\| \le \|\mathbf{t} - \mathbf{y}\| \sqrt{1 - 1/2n}$, which would imply that $(\mathbf{B}, \mathbf{t}')$ is an instance of $\mathrm{BDD}'_{\alpha\sqrt{1-1/2n}}$. And therefore solving polynomially many (we need to find a d in the correct range as well as try all $2n$ possibilities in the set S) instances of $\mathrm{BDD}'_{\alpha\sqrt{1-1/2n}}$ would result in a solution to BDD'_{α}.

Without loss of generality we can assume that $\mathbf{y} = \mathbf{0}$. Then $\|\mathbf{t}\|^2 = \sum_i t_i^2 = D^2$, and there must exist an i such that $|t_i| \ge \frac{D}{\sqrt{n}}$. Then for a $j \in \{-1, 1\}$ that has the same sign as t_i, the vector $\mathbf{t}' = \mathbf{t} - \frac{jd}{\sqrt{n}} \mathbf{u}_i$ is in S and

$$\begin{aligned} \|\mathbf{t}'\|^2 &= \|(t_1, \ldots, t_{i-1}, t_i - \frac{jd}{\sqrt{n}}, t_{i+1}, \ldots, t_n)\|^2 \\ &= D^2 - \frac{2|t_i|d}{\sqrt{n}} + \frac{d^2}{n} \le D^2 - \frac{2dD}{n} + \frac{d^2}{n} \le D^2 \left(1 - \frac{1}{2n}\right) \end{aligned}$$

where the last inequality follows from (1). □

Notice that the above lemma cannot be combined with itself to obtain a reduction from BDD'_{α} to BDD'_{β} for an arbitrarily small β. This is because if $\beta = \alpha \left(\sqrt{1 - 1/2n}\right)^c$, we would need to solve $poly(n)^c$ instances of BDD'_{β} in order to solve one instance of BDD'_{α}. This is doable in polynomial time only if c is a constant, which leads to the following corollary:

Corollary 1. *For any $\alpha \ge 1$ and any constant c, there is a polynomial-time Cook reduction from BDD'_{α} to $\mathrm{BDD}'_{\alpha(1-1/n)^c}$.*

Combining the above corollary with Lemma 2, we obtain:

Corollary 2. *For any $\alpha \ge 1$ and any constant c, there is a polynomial-time Cook reduction from BDD_{α} to $\mathrm{BDD}_{\alpha(1-1/n)^c}$.*

4 Reducing BDD to uSVP

In this section we present the reduction from from the BDD problem to uSVP. Given an instance $(\mathbf{B}, \mathbf{t})$ of BDD, we construct a uSVP instance as in (2), where μ is the approximate distance from $\mathbf{t}$ to $\mathcal{L}(\mathbf{B})$ (we do not know this distance, but can guess a good-enough approximation). The idea is that if $(\mathbf{B}, \mathbf{t})$ is an instance of $\mathrm{BDD}_{1/(2\gamma)}$ for $\gamma \ge 1$, then the lattice $\mathcal{L}(\mathbf{B}')$ has a γ-unique shortest vector and this vector is formed by using the last column of $\mathbf{B}'$ exactly once. Therefore finding this shortest vector allows us to find the closest vector in $\mathcal{L}(\mathbf{B})$ to $\mathbf{t}$.

Theorem 1. *For any $\gamma \geq 1$, there is a polynomial time Cook-reduction from* $\mathrm{BDD}_{1/(2\gamma)}$ *to* uSVP_γ.

Proof. Let $(\mathbf{B}, \mathbf{t})$ be an instance of $\mathrm{BDD}_{1/(2\gamma)}$ and let $\mu = \mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) < \lambda_1(\mathbf{B})/(2\gamma)$. Let $\mathbf{v}$ be a vector in $\mathcal{L}(\mathbf{B})$ such that $\|\mathbf{t} - \mathbf{v}\| = \mu$. The goal of the reduction is to use a uSVP_γ oracle to find $\mathbf{v}$. For simplicity, we will assume that μ is known (we will explain how to deal with this issue at the end of the proof), and define the matrix

$$\mathbf{B}' = \begin{bmatrix} \mathbf{B} & \mathbf{t} \\ \mathbf{0}^T & \mu \end{bmatrix} \tag{2}$$

We will show that the lattice $\mathcal{L}(\mathbf{B}')$ contains a γ-unique shortest vector $\mathbf{v}' = [(\mathbf{v} - \mathbf{t})^T, -\mu]^T$, and therefore finding such a vector will recover the vector $\mathbf{v}$, which is the solution to the BDD instance. The length of $\mathbf{v}'$ is $\sqrt{\mu^2 + \mu^2} = \sqrt{2}\mu$ and so we need to show that all other vectors in $\mathcal{L}(\mathbf{B}')$ that are not multiples of $\mathbf{v}'$ have length at least $\lambda_1(\mathbf{B})/\sqrt{2} > \sqrt{2}\gamma\mu$.

Assume for the sake of contradiction that $\mathbf{w}'$ is a vector in $\mathcal{L}(\mathbf{B}')$ of length less than $\lambda_1(\mathbf{B})/\sqrt{2}$ that is not a multiple of $\mathbf{v}'$. We can rewrite the vector $\mathbf{w}' = [(\mathbf{w} - \beta\mathbf{t})^T, -\beta\mu]^T$ where $\beta \geq 0$ and $\mathbf{w} \in \mathcal{L}(\mathbf{B})$, and so

$$\frac{\lambda_1(\mathbf{B})}{\sqrt{2}} > \|\mathbf{w}'\| = \sqrt{\|\mathbf{w} - \beta\mathbf{t}\|^2 + (\beta\mu)^2},$$

which implies that $\beta\mu < \lambda_1(\mathbf{B})/\sqrt{2}$ and

$$\|\mathbf{w} - \beta\mathbf{t}\| < \sqrt{\frac{\lambda_1(\mathbf{B})^2}{2} - (\beta\mu)^2}.$$

Now consider the vector $\mathbf{w} - \beta\mathbf{v} \in \mathcal{L}(\mathbf{B})$. Since we assumed that $\mathbf{w}'$ was not a multiple of $\mathbf{v}'$, the vector $\mathbf{w} - \beta\mathbf{v}$ is a non-zero lattice vector. To get the contradiction, we will show that the length of this vector is strictly less than $\lambda_1(\mathbf{B})$. Using the triangular inequality, we rewrite

$$\|\mathbf{w} - \beta\mathbf{v}\| = \|\mathbf{w} - \beta\mathbf{t} - \beta(\mathbf{v} - \mathbf{t})\| \leq \|\mathbf{w} - \beta\mathbf{t}\| + \beta\|\mathbf{v} - \mathbf{t}\| < \sqrt{\frac{\lambda_1(\mathbf{B})^2}{2} - (\beta\mu)^2} + \beta\mu.$$

The last term of the above inequality is maximized when $\beta = \lambda_1(\mathbf{B})/(2\mu)$, and therefore for all β,

$$\|\mathbf{w} - \beta\mathbf{v}\| < \sqrt{\frac{\lambda_1(\mathbf{B})^2}{2} - (\beta\mu)^2} + \beta\mu \leq \lambda_1(\mathbf{B}),$$

which gives us the contradiction.

We now discuss the issue of guessing the μ such that $\mu = \mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$. While we cannot guess such a μ exactly, we can, in polynomial time, guess a μ such that $(1 - 1/n)\mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq \mu \leq (1 + 1/n)\mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$. We can do this because we can find a d such that $\mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq d \leq 2^n\mathrm{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B}))$

(Lemma 3), and so trying all the possible values of μ in the polynomial-sized set $\{d(1+1/n)^i : 0 \le i \le \log_{1+1/n} 2^n\}$ at least one "good" μ. We can then redo the above proof by appropriately modifying some terms by factors of $1-1/n$ or $1+1/n$ in order to satisfy the inequalities that appear. The end result will be that we will have a reduction not from $\mathrm{BDD}_{1/(2\gamma)}$, but from the slightly easier $\mathrm{BDD}_{(1-1/n)^c/(2\gamma)}$ problem for some small constant c. But we can then apply Corollary 2 to obtain the claimed reduction from $\mathrm{BDD}_{1/(2\gamma)}$ to uSVP_γ. □

5 Reducing uSVP to BDD

Theorem 2. *For any polynomially bounded $\gamma(n) = n^{O(1)}$, there is a polynomial time Cook-reduction from uSVP_γ to $\mathrm{BDD}_{1/\gamma}$.*

Proof. Let $\mathbf{B}$ be a uSVP_γ instance, i.e., an n-dimensional lattice such that $\lambda_2(\mathbf{B}) > \gamma\lambda_1(\mathbf{B})$, and let p be the smallest prime bigger than $\gamma(n)$. (Since $\gamma(n)$ is polynomially bounded, such a prime can be easily found using trial division.) We want to find the shortest nonzero vector in $\mathcal{L}(\mathbf{B})$. We proceed similarly to the reduction from SVP to CVP of Goldreich, Micciancio, Safra and Seifert. (The GMSS reduction corresponds to the special case when $p = 2$.) For any $i = 1, \ldots, n$, we consider the lattice

$$\mathbf{B}^{(i)} = [\mathbf{b}_1, \ldots, \mathbf{b}_{i-1}, p\mathbf{b}_i, \mathbf{b}_{i+1}, \ldots, \mathbf{b}_n]$$

and invoke the BDD oracle on input $(\mathbf{B}^{(i)}, j \cdot \mathbf{b}_i)$ for $j = 1, \ldots, p-1$. Assume without loss of generality that the oracle always returns a lattice vector. (If the input instance violates the BDD promise, the oracle may simply return $\mathbf{0}$.) Let $\mathbf{v}_{i,j} \in \mathcal{L}(\mathbf{B}^{(i)}) \subset \mathcal{L}(\mathbf{B})$ the lattice vector returned by the oracle on input $(\mathbf{B}^{(i)}, j \cdot \mathbf{b}_i)$, and let

$$\mathbf{w}_{i,j} = \mathbf{v}_{i,j} - j \cdot \mathbf{b}_i.$$

Notice that all vectors $\mathbf{w}_{i,j}$ belong to the lattice $\mathcal{L}(\mathbf{B})$ because $\mathbf{v}_{i,j} \in \mathcal{L}(\mathbf{B})$ and $\mathbf{b}_i \in \mathcal{L}(\mathbf{B})$. The reduction outputs the smallest nonzero vector among the $\mathbf{w}_{i,j}$.

In order to prove the reduction correct, we need to show that at least one of the $\mathbf{w}_{i,j}$ has length $\lambda_1(\mathbf{B})$, so that the reduction outputs a shortest nonzero vector in $\mathcal{L}(\mathbf{B})$. Let $\mathbf{u} = \mathbf{B}\mathbf{x}$ be the shortest nonzero vector in $\mathcal{L}(\mathbf{B})$. Clearly, there must exists an $i \in \{1, \ldots, n\}$ such that p does not divide x_i, because otherwise $\mathbf{u}/p = \mathbf{B}(\mathbf{x}/p) \in \mathcal{L}(\mathbf{B})$ is an even shorter nonzero lattice vector. Fix this i, and let $j = (-x_i \bmod p) \in \{1, \ldots, p-1\}$. We claim that $(\mathbf{B}^{(i)}, j \cdot \mathbf{b}_i)$ is a valid $\mathrm{BDD}_{1/\gamma}$ instance and $\mathrm{dist}(\mathbf{B}^{(i)}, j \cdot \mathbf{b}_i) = \lambda_1(\mathbf{B})$. It will follow that on input $(\mathbf{B}^{(i)}, j \cdot \mathbf{b}_i)$, the BDD oracle returns the lattice vector $\mathbf{v}_{i,j} \in \mathcal{L}(\mathbf{B}^{(i)})$ closest to $j \cdot \mathbf{b}_i$, and

$$\|\mathbf{w}_{i,j}\| = \|\mathbf{v}_{i,j} - j\mathbf{b}_i\| = \mathrm{dist}(\mathbf{B}^{(i)}, j\mathbf{b}_i) = \lambda_1(\mathbf{B}).$$

First, notice that $j\mathbf{b}_i$ is within distance $\|\mathbf{u}\| = \lambda_1(\mathbf{B})$ from $\mathcal{L}(\mathbf{B}^{(i)})$ because

$$\mathbf{u} = \sum_{k=1}^{n} \mathbf{b}_k x_k = \sum_{k \neq i} \mathbf{b}_k x_k + \mathbf{b}_i(x_i + j) - j\mathbf{b}_i$$

and $(x_i + j)$ is a multiple of p. Moreover, $j\mathbf{b}_i \notin \mathcal{L}(\mathbf{B}^{(i)})$ because any vector in $\mathcal{L}(\mathbf{B}^{(i)}) - j\mathbf{b}_i$ uses $\mathbf{b}_i$ a nonzero (modulo p) number of times. Therefore, $\mathrm{dist}(j\mathbf{b}_i, \mathcal{L}(\mathbf{B}^{(i)})) = \lambda_1(\mathbf{B})$. We also need to show that $(\mathbf{B}^{(i)}, j\mathbf{b}_i)$ is a valid $\mathrm{BDD}_{1/\gamma}$ instance, i.e., $\mathrm{dist}(\mathbf{B}^{(i)}, j\mathbf{b}_i) < (1/\gamma)\lambda_1(\mathbf{B}^{(i)})$, or, equivalently, $\lambda_1(\mathbf{B}^{(i)}) > \gamma\lambda_1(\mathbf{B})$. To this end, consider any nonzero vector $\mathbf{y} = \mathbf{B}^{(i)}\mathbf{z}$. If $\mathbf{y}$ is linearly independent from $\mathbf{u}$, then we immediately get $\|\mathbf{y}\| \geq \lambda_2(\mathbf{B}) > \gamma\lambda_1(\mathbf{B})$. So, assume $\mathbf{y} = c\mathbf{u}$ for some $c \in \mathbb{Z}\backslash\{0\}$. Using the definition of $\mathbf{B}^{(i)}$ and the equality $\mathbf{B}^{(i)}\mathbf{z} = c\mathbf{B}\mathbf{x}$, we get $pz_i = cx_i$ (and $z_k = cx_k$ for all $k \neq i$). Since p does not divide x_i (by our choice of i), p must divide c, and $\|\mathbf{y}\| = c\|\mathbf{u}\| \geq p\|\mathbf{u}\| > \gamma\lambda_1(\mathbf{B})$. □

6 Reducing uSVP to GapSVP

Theorem 3. *For any polynomially bounded γ, given an oracle for* $\textsc{GapSVP}_\gamma$*, we can solve* $\textsc{uSVP}_\gamma$*. Moreover, all calls to the* $\textsc{GapSVP}_\gamma$ *oracle are of the form* $(\mathbf{B}, d)$ *where* $\lambda_2(\mathbf{B}) > \gamma d$*.*

Proof. Let $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k]$ be the basis of a lattice satisfying $\lambda_2(\mathbf{B}) > \gamma\lambda_1(\mathbf{B})$, and let $\mathbf{u}$ be the (unique) shortest vector in $\mathcal{L}(\mathbf{B})$. Without loss of generality we assume $\mathbf{B}$ is an integer lattice. We show how to use the $\textsc{GapSVP}_\gamma$ oracle to obtain a lower rank sublattice of $\mathcal{L}(\mathbf{B}')$ that still contains the lattice vector $\mathbf{u}$ of length $\lambda = \lambda_1(\mathbf{B})$. The shortest vector in $\mathcal{L}(\mathbf{B})$ can then be found by iteratively applying this procedure, until the rank of the lattice is reduced to 1, and $\mathbf{B}' = [\pm\mathbf{u}]$.

In fact it is enough to show how to find any full-rank proper sublattice $\mathcal{L}(\mathbf{B}') \subset \mathcal{L}(\mathbf{B})$ still containing $\mathbf{u}$. If we repeat this $t > n(n + \log_2 n)$ times, the result will be a sublattice $\mathbf{S}$ such that $\det(\mathbf{S}) \geq 2^t \det(\mathbf{B})$, because each time we select a sublattice the value of the determinant at least doubles. The dual $\mathbf{D}$ of this sublattice will have determinant $\det(\mathbf{D}) \leq 1/(2^t \det(\mathbf{B}))$, and using the LLL algorithm we can find a dual vector $\mathbf{v} \in \mathcal{L}(\mathbf{D})$ of length

$$\|\mathbf{v}\| \leq 2^n \sqrt{n} \det(\mathbf{D})^{1/n} \leq \frac{\sqrt{n}2^n}{2^{t/n} \det(\mathbf{B})^{1/n}}.$$

By Minkowski's bound we have $\|\mathbf{u}\| \leq \sqrt{n} \det(\mathbf{B})^{1/n}$ and therefore by the Cauchy-Schwarz inequality,

$$|\langle \mathbf{u}, \mathbf{v} \rangle| \leq \|\mathbf{v}\| \cdot \|\mathbf{u}\| \leq n 2^{n - t/n} < 1.$$

But $\langle \mathbf{u}, \mathbf{v} \rangle$ is an integer because $\mathbf{u} \in \mathcal{L}(\mathbf{S})$ and $\mathbf{v} \in \mathcal{L}(\mathbf{D})$ and the lattices $\mathcal{L}(\mathbf{S})$ and $\mathcal{L}(\mathbf{D})$ are dual. So, it must be $\langle \mathbf{u}, \mathbf{v} \rangle = 0$, i.e., $\mathbf{u}$ is orthogonal to $\mathbf{v}$. Taking the sublattice of $\mathbf{S}$ orthogonal to $\mathbf{v}$ gives a lower rank sublattice $\mathcal{L}(\mathbf{B}') \subset \mathcal{L}(\mathbf{B})$ still containing $\mathbf{u}$.

So, all we need to do is to show that the $\textsc{GapSVP}$ oracle can be used to find a proper sublattice $\mathcal{L}(\mathbf{B}') \subset \mathcal{L}(\mathbf{B})$ that still contains $\mathbf{u}$. Let p be a prime bigger than γ and consider the sublattices $\mathbf{B}_0 = [p\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_k]$ and $\mathbf{B}_c =$

$[\mathbf{b}_1 + c\mathbf{b}_2, p\mathbf{b}_2, \mathbf{b}_3, \ldots, \mathbf{b}_k]$ for $c = 1, \ldots, p$. We claim that there exists a c such that $\mathbf{u} \in \mathcal{L}(\mathbf{B}_c)$. Moreover, for any c, if $\mathbf{u} \notin \mathcal{L}(\mathbf{B}_c)$, then

$$\lambda_1(\mathbf{B}_c) \geq \min(\lambda_2(\mathbf{B}), p\lambda_1(\mathbf{B})) > \gamma\lambda.$$

In other words, the instances $(\mathbf{B}_c, \lambda)$, will always fulfill the promise that either $\lambda_1(\mathbf{B}_c) \leq \lambda$ or $\lambda_1(\mathbf{B}_c) > \gamma\lambda$. So, if we could invoke the GAPSVP$_\gamma$ oracle on inputs $(\mathbf{B}_c, \lambda)$ for $c = 0, \ldots, p$, then the oracle would output YES for at least some c, and for any such c we would have $\mathcal{L}(\mathbf{B}_c) \leq \lambda$. However, we cannot make these oracle calls because the value λ is not known, and also because λ might be an irrational number. Both problems can be easily solved by performing a binary search as follows. Compute an approximation d to λ using a lattice approximation algorithm. Say, $\lambda \leq d < 2^n\lambda$. If we invoke the oracle on inputs $(\mathbf{B}_c, d)$, then the oracle will output YES for at least some $c \in \{0, \ldots, p\}$. On the other hand, if we invoke the oracle in inputs $(\mathbf{B}, d/2^n)$, then the oracle will output NO for all c because $d/2^n < \lambda \leq \lambda_1(\mathbf{B}_c)$. Using binary search we can find a $d' < d''$ in $[d/2^n, d)$ such that

- on input $(\mathbf{B}_c, d')$ the oracle outputs NO for all c,
- on input $(\mathbf{B}_c, d'')$ the oracle outputs YES for some c,
- $d'' - d' < 1/(2\gamma^2 d)$.

Notice that the number of iterations performed by the binary search procedure is at most $\log_2(2^n d) + \log_2(2\gamma^2 d) \leq n + 1 + 2\log_2(\gamma d)$ which is polynomial in the input size. From the condition $d'' - d' < 1/(2\gamma^2 d)$, we get that the interval $[(\gamma d')^2, (\gamma d'')^2]$ contains at most one integer because

$$(\gamma d'')^2 - (\gamma d')^2 = \gamma^2(d'' - d')(d'' + d') < 1.$$

Similarly, $(d'')^2 - (d')^2 < 1$ and $[(d')^2, (d'')^2]$ also contains at most one integer. We know that $d' < \lambda$ because the oracle outputs NO on all queries $(\mathbf{B}_c, d')$. Since $\mathbf{B}$ is an integer lattice, λ^2 is an integer. If $[(d')^2, (d'')^2]$ contains no integer value, then it must be $\lambda > d''$, and for all oracle calls $(\mathbf{B}_c, d'')$ that were answered with YES, it must be $\lambda_1(\mathbf{B}_c) \leq \lambda$. On the other hand, if $[(d')^2, (d'')^2]$ contains an integer k, it may or may not be the case that $\lambda = \sqrt{k}$. There are two cases:

- If $(\gamma^2 k, (\gamma d'')^2]$ contains no integer, then for every c, either $\lambda_1(\mathbf{B}_c) \leq d''$ or $\lambda_1(\mathbf{B}_c) > \gamma d''$. So, we can proceed as before, and select any c for which the oracle output YES on input $(\mathbf{B}_c, d'')$.
- If there is an integer $k' \in (\gamma^2 k, (\gamma d'')^2]$, then we select any value $d_0 \in [\sqrt{k}, \sqrt{k'}/\gamma)$, and call the oracle again on input $(\mathbf{B}_c, d_0)$. The oracle will output YES on at least one of these calls, and the corresponding lattice is guaranteed to satisfy $\lambda_1(\mathbf{B}_c) \leq \lambda$.

We will now prove the claim that there exists a $c \in \{0, \ldots, p\}$ such that $\mathbf{u} \in \mathcal{L}(\mathbf{B}_c)$. Let $\mathbf{u} = \mathbf{B}\mathbf{x}$ for some integer vector $\mathbf{x} = (x_1, x_2, \ldots, x_k)^T$. If $p \mid x_1$, then clearly $\mathbf{u}$ is a vector in $\mathcal{L}(\mathbf{B}_0)$. If $p \nmid x_1$, then we will show that $\mathbf{u} \in \mathcal{L}(\mathbf{B}_c)$ for $c = x_2 x_1^{-1} (\bmod\, p)$. Consider the vector $\mathbf{x}' = (x_1, (x_2 - cx_1)/p, x_3, \ldots, x_n)$. Notice that $\mathbf{B}_c\mathbf{x}' = \mathbf{B}\mathbf{x}$ and by our choice of c, $\mathbf{x}$ has all integer coordinates since $x_2 - cx_1 \equiv 0 (\bmod\, p)$. Therefore $\mathbf{u}$ is also a vector in $\mathcal{L}(\mathbf{B}_c)$. □

7 Reducing GapSVP to BDD

In this section we give a reduction from GapSVP to BDD. When combined with the BDD to uSVP reduction from section 4, we obtain the GapSVP to uSVP reduction which proves that the Ajtai-Dwork [2] and the Regev [33] cryptosystems are based on the hardness of the approximate minimum distance problem. As mentioned earlier, the GapSVP to BDD reduction is already implicit in the recent work of Peikert [32]. We repeat it here for completeness and also because in Peikert's work, this reduction is entangled with some extra technicalities that pertain to his main result.

Theorem 4. *For any $\gamma > 2\sqrt{n/\log n}$ there is a polynomial time Cook-reduction from* GapSVP_γ *to* $\text{BDD}_{\frac{1}{\gamma}\sqrt{n/\log n}}$.

Proof. Let $(\mathbf{B}, d)$ be an instance of GapSVP_γ. We need to output YES if $\lambda_1(\mathbf{B}) \leq d$ and NO if $\lambda_1(\mathbf{B}) > \gamma d$. In all other instances, any answer will suffice.

We repeat the following procedure *poly*(n) times. Generate a uniformly random point $\mathbf{s}$ in $\mathcal{B}(0, d\sqrt{n/\log n})$, and let $\mathbf{t} = \mathbf{s} \bmod \mathbf{B}$. Feed the instance $(\mathbf{B}, \mathbf{t})$ to the $\text{BDD}_{\frac{1}{\gamma}\sqrt{n/\log n}}$ oracle and receive the answer $\mathbf{v}$. If we ever have the case that $\mathbf{v} \neq \mathbf{t} - \mathbf{s}$, we output YES. On the other hand, if all *poly*(n) calls to the oracle result in $\mathbf{v}$'s such that $\mathbf{v} = \mathbf{t} - \mathbf{s}$, we output NO.

We will now prove that the reduction is correct. Suppose that $(\mathbf{B}, d)$ is a NO instance of GapSVP_γ. Then

$$\text{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) = \text{dist}(\mathbf{s}, \mathcal{L}(\mathbf{B})) \leq d\sqrt{n/\log n} < \frac{\lambda_1(\mathbf{B})}{\gamma}\sqrt{n/\log n},$$

and so $(\mathbf{B}, \mathbf{t})$ is a valid instance of $\text{BDD}_{\frac{1}{\gamma}\sqrt{n/\log n}}$. Furthermore, since $\gamma > 2\sqrt{n/\log n}$, the distance of $\mathbf{t}$ from the lattice is less than $\lambda_1(\mathbf{B})/2$, and so there is only one possible lattice vector within distance $\frac{\lambda_1(\mathbf{B})}{\gamma}\sqrt{n/\log n}$ of $\mathbf{t}$. And since the lattice vector $\mathbf{v} = \mathbf{t} - \mathbf{s}$ is at a distance $\|\mathbf{s}\| \leq d\sqrt{n/\log n} < \frac{\lambda_1(\mathbf{B})}{\gamma}\sqrt{n/\log n}$ away from $\mathbf{t}$, it must be the vector that the BDD oracle returns. So, the reduction certainly outputs NO.

Now suppose that that $(\mathbf{B}, d)$ is a YES instance of GapSVP_γ, which means that $\lambda_1(\mathbf{B}) \leq d$. Let $\mathbf{x}$ be a lattice point whose length is $\lambda_1(\mathbf{B})$. In order for the BDD oracle to successfully fool us into replying NO, he needs to output $\mathbf{v} = \mathbf{t} - \mathbf{s}$ in every round of the protocol. Notice that this is equivalent to the oracle knowing $\mathbf{s}$. But every time we pick an $\mathbf{s}$ and reveal $\mathbf{t} = \mathbf{s} \bmod \mathbf{B}$ to the oracle, by Lemma 4, there is some $1/$*poly*(n) probability δ that $\|\mathbf{s} - \mathbf{x}\| \leq d\sqrt{n/\log n}$. And in this case, given $\mathbf{t}$, the oracle cannot know with probability greater than 1/2 whether we randomly generated $\mathbf{s}$ or $\mathbf{s} - \mathbf{x}$ (since both $\mathbf{s} \bmod \mathbf{B}$ and $\mathbf{s} - \mathbf{x} \bmod \mathbf{B}$ equal to $\mathbf{t}$). Therefore for a δ fraction of the $\mathbf{t}$'s that we give him, the oracle cannot guess the exact $\mathbf{s}$ with probability greater than 1/2. And so guessing the $\mathbf{s}$ in all $\text{poly}(n) = n/\delta$ rounds has negligible success probability. Therefore with probability exponentially close to 1, some $\mathbf{v}$ will not equal $\mathbf{t} - \mathbf{s}$ and our algorithm will reply YES. □

8 Reductions for Other ℓ_p Norms

Throughout this work, we have only dealt with the ℓ_2 norm, and we now briefly discuss how our reductions translate to arbitrary ℓ_p norms. The reduction from uSVP to BDD and uSVP to GapSVP in sections 5 and 6 don't rely on any specific properties of the ℓ_2 norm and so the reductions go through for other norms with only very slight modifications. In the reduction from BDD to uSVP in section 4, we repeatedly used the definition of the ℓ_2 norm, and so the reductions do not go straight through. Nevertheless, a simple modification of the proof which involves appropriately changing the equalities and inequalities to correspond with the definitions of the ℓ_p norm of interest, results in a reduction from $\mathrm{BDD}_{1/(2\gamma)}$ to uSVP_γ just as for the ℓ_2 norm.

The only reduction that becomes weaker for ℓ_p norms where $p \neq 2$ is the reduction from GapSVP to BDD in section 7. The $\sqrt{n/\log n}$ factor loss in the reduction for the ℓ_2 norm is directly tied to the fact that spheres that are a distance of d apart must have radii of at least $d\sqrt{n/\log n}$ in order for their intersecting volume to be a non-negligible fraction of their total volume (Lemma 4). On the other hand, in ℓ_p norms for $p \neq 2$, the radii of the spheres have to be larger for their intersection to be non-negligible. It's not hard to see that for the ℓ_1 and ℓ_∞ norms, the radii need to be at least $dn/\log n$, and it is shown in [9] that this suffices for all other ℓ_p norms as well (although it is not a tight bound when $1 < p < \infty$). So essentially using an analogue of Lemma 4 for other ℓ_p norms, we can obtain a reduction from GapSVP_γ to $\mathrm{BDD}_{\frac{1}{\gamma}n/logn}$ for any $\gamma > 2n/\log n$.

Acknowledgements We thank the anonymous referees for very useful comments.

References

1. M. Ajtai. Generating hard instances of lattice problems. In *STOC*, pages 99–108, 1996.
2. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, 1997. An improved version is described in ECCC 2007.
3. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *STOC*, pages 601–610, 2001.
4. L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.
5. J. Y. Cai. A relation of primal-dual lattices and the complexity of shortest lattice vector problem. *Theor. Comput. Sci.*, 207(1):105–116, 1998.
6. J. Y. Cai. On the average-case hardness of CVP. In *FOCS*, pages 308–317, 2001.
7. N. Gama and P. Q. Nguyen. Predicting lattice reduction. In *EUROCRYPT*, 2008.
8. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices, and new cryptographic constructions. In *STOC*, 2008.
9. O. Goldreich and S. Goldwasser. On the limits of nonapproximability of lattice problems. *J. Comput. Syst. Sci.*, 60(3):540–563, 2000.
10. O. Goldreich, S. Goldwasser, and S. Halevi. Collision-free hashing from lattice problems. Electronic Colloquium on Computational Complexity (ECCC), 1996.

11. O. Goldreich, S. Goldwasser, and S. Halevi. Eliminating decryption errors in the Ajtai-Dwork cryptosystem. In *CRYPTO*, pages 105–111, 1997.
12. O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *CRYPTO*, pages 112–131, 1997.
13. O. Goldreich, D. Micciancio, S. Safra, and J.P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.
14. I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *STOC*, pages 469–477, 2007.
15. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *ANTS*, pages 267–288, 1998.
16. R. Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147, 1995.
17. R. Kannan. Algorithmic geometry of numbers. *Annual Review of Computer Science*, 2:231–267, 1987.
18. A. Kawachi, K. Tanaka, and K. Xagawa. Concurrently secure identification schemes based on the worst-case hardness of lattice problems. In *ASIACRYPT*, 2008.
19. S. Khot. Hardness of approximating the shortest vector problem in lattices. In *FOCS*, pages 126–135, 2004.
20. A. Klivans and A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *FOCS*, pages 553–562, 2006.
21. R. Kumar and D. Sivakumar. On the unique shortest lattice vector problem. *Theor. Comput. Sci.*, 255(1-2):641–648, 2001.
22. J. C. Lagarias and A. M. Odlyzko. Solving low density subset sum problems. *Journal of the ACM*, 32:229–246, 1985.
23. A. K. Lenstra, H. W. Lenstra Jr., and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, (261):513–534, 1982.
24. Y.-K. Liu, V. Lyubashevsky, and D. Micciancio. On bounded distance decoding for general lattices. In *RANDOM*, 2006.
25. V. Lyubashevsky. Lattice-based identification schemes secure under active attacks. In *Public Key Cryptography*, pages 162–179, 2008.
26. V. Lyubashevsky and D. Micciancio. Asymptotically efficient lattice-based digital signatures. In *TCC*, pages 37–54, 2008.
27. D. Micciancio. Efficient reductions among lattice problems. In *SODA*, pages 84–93, 2008.
28. D. Micciancio and S. Goldwasser. *Complexity Of Lattice Problems: A Cryptographic Perspective*. Kluwer Academic Publishers, 2002.
29. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. on Computing*, 37(1):267–302, 2007.
30. D. Micciancio and S. Vadhan. Statistical zero-knowledge proofs with efficient provers: Lattice problems and more. In *CRYPTO*, pages 282–298, 2003.
31. P. Q. Nguyen. Cryptanalysis of the Goldreich-Goldwasser-Halevi cryptosystem from Crypto '97. In *CRYPTO*, pages 288–304, 1999.
32. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem. In *STOC*, 2009.
33. O. Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, 2004.
34. O. Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, 2004.
35. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, 2005.